G13AGF – NAG Fortran Library Routine Document

Note. Before using this routine, please read the Users' Note for your implementation to check the interpretation of bold italicised terms and other implementation-dependent details.

1 Purpose

G13AGF accepts a series of new observations of a time series, the model of which is already fully specified, and updates the 'state set' information for use in constructing further forecasts. The previous specifications of the time series model should have been obtained by using G13AEF or G13AFF to estimate the relevant parameters. The supplied state set will originally have been produced by G13AEF or G13AFF, but may since have been updated by earlier calls to G13AGF.

A set of residuals corresponding to the new observations is returned. These may be of use in checking that the new observations conform to the previously fitted model.

2 Specification

```
    SUBROUTINE G13AGF(ST, NST, MR, PAR, NPAR, C, ANX, NUV, ANEXR, WA,
   1                   NWA, IFAIL)
    INTEGER            NST, MR(7), NPAR, NUV, NWA, IFAIL
    real               ST(NST), PAR(NPAR), C, ANX(NUV), ANEXR(NUV),
   1                   WA(NWA)
```

3 Description

The time series model is specified as outlined in Section 3 of the document for G13AEF or Section 3 of the document for G13AFF. This also describes how the state set, which contains the minimum amount of time series information needed to construct forecasts, is made up of

(i)   the differenced series w_t (uncorrected for the constant c), for $(N - P \times s) < t \le N$,

(ii)   the d' values required to reconstitute the original series x_t from the differenced series w_t,

(iii)   the intermediate series e_t, for $(N - \max(p, Q \times s)) < t \le N$,

(iv)   the residual series a_t, for $(N - q) < t \le N$.

If the number of original undifferenced observations was n, then $d' = d + (D \times s)$ and $N = n - d'$.

To update the state set, given a number of new undifferenced observations x_t, $t = n+1, n+2, \ldots, n+k$, the four series above are first reconstituted.

Differencing and residual calculation operations are then applied to the new observations and k new values of w_t, e_t and a_t are derived.

The first k values in these three series are then discarded and a new state set is obtained.

The residuals in the a_t series corresponding to the k new observations are preserved in an output array. The parameters of the time series model are not changed in this routine.

4 References

None.

5 Parameters

1:   ST(NST) — *real* array   *Input/Output*

On entry: the state set derived from G13AEF or G13AFF, or as modified using earlier calls of G13AGF.

On exit: the updated values of the state set.

2:   NST — INTEGER   *Input*

   On entry: the number of values in the state set array ST.

   Constraint: $\text{NST} = P \times s + D \times s + d + q + \max(p, Q \times s)$. (As returned by G13AEF or G13AFF).

3:   MR(7) — INTEGER array   *Input*

   On entry: the orders vector (p, d, q, P, D, Q, s) of the ARIMA model, in the usual notation.

   Constraints:

   $$p, d, q, P, D, Q, s \geq 0,$$
   $$p + q + P + Q > 0,$$
   $$s \neq 1,$$
   $$\text{if } s = 0, \text{ then } P + D + Q = 0,$$
   $$\text{if } s > 1, \text{ then } P + D + Q > 0.$$

4:   PAR(NPAR) — *real* array   *Input*

   On entry: the estimates of the p values of the ϕ parameters, the q values of the θ parameters, the P values of the Φ parameters and the Q values of the Θ parameters in the model – in that order, using the usual notation.

5:   NPAR — INTEGER   *Input*

   On entry: the number of ϕ, θ, Φ, Θ parameters in the model.

   Constraint: $\text{NPAR} = p + q + P + Q$.

6:   C — *real*   *Input*

   On entry: the constant to be subtracted from the differenced data.

7:   ANX(NUV) — *real* array   *Input*

   On entry: the new undifferenced observations which are to be used to update ST.

8:   NUV — INTEGER   *Input*

   On entry: the number of new observations in ANX, k.

9:   ANEXR(NUV) — *real* array   *Output*

   On exit: the residuals corresponding to the new observations in ANX.

10:   WA(NWA) — *real* array   *Workspace*
11:   NWA — INTEGER   *Input*

   On entry: the dimension of the array WA as declared in the (sub)program from which G13AGF is called.

   Constraint: $\text{NWA} \geq (4 \times \text{NPAR} + 3 \times \text{NST})$.

12:   IFAIL — INTEGER   *Input/Output*

   On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

   On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

On entry, $\text{NPAR} \neq p + q + P + Q$,

or the orders vector MR is invalid (check the constraints in Section 5).

IFAIL = 2

On entry, $\text{NST} \neq P \times s + D \times s + d + q + \max(Q \times s, p)$.

IFAIL = 3

On entry, $\text{NUV} \leq 0$.

IFAIL = 4

On entry, $\text{NWA} < 4 \times \text{NPAR} + 3 \times \text{NST}$.

7 Accuracy

The computations are believed to be stable.

8 Further Comments

The time taken by the routine is approximately proportional to $\text{NUV} \times \text{NPAR}$.

9 Example

The following program is based on data derived from a study of monthly airline passenger totals (in thousands) to which a logarithmic transformation had been applied. The time series model was based on seasonal and non-seasonal differencing both of order 1, with seasonal period 12. The number of parameters estimated was two; a non-seasonal moving average parameter θ_1 with value 0.327 and a seasonal moving average parameter Θ_1 with value 0.6270. There was no constant correction. These, together with the state set array, were obtained using G13AEF.

Twelve new observations are supplied. The routine updates the state set and outputs a set of residuals corresponding to the new observations.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     G13AGF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER           NPMAX, NSTMAX, NNVMAX, NWAMAX
      PARAMETER         (NPMAX=10,NSTMAX=50,NNVMAX=50,NWAMAX=150)
      INTEGER           NIN, NOUT
      PARAMETER         (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real              C
      INTEGER           I, IFAIL, NNV, NPAR, NST, NWA
*     .. Local Arrays ..
      real              ANV(NNVMAX), ANVR(NNVMAX), PAR(NPMAX),
     +                  ST(NSTMAX), WA(NWAMAX)
      INTEGER           MR(7)
```

```
*       .. External Subroutines ..
        EXTERNAL          G13AGF
*       .. Intrinsic Functions ..
        INTRINSIC         MAX
*       .. Executable Statements ..
        WRITE (NOUT,*) 'G13AGF Example Program Results'
*       Skip heading in data file
        READ (NIN,*)
        READ (NIN,*) NNV
        READ (NIN,*) (MR(I),I=1,7)
        NPAR = MR(1) + MR(3) + MR(4) + MR(6)
        NST = MR(4)*MR(7) + MR(5)*MR(7) + MR(2) + MR(3) + MAX(MR(1),MR(6)
     +        *MR(7))
        NWA = 4*NPAR + 3*NST
        IF (NNV.GT.0 .AND. NNV.LE.NNVMAX .AND. NPAR.GT.0 .AND. NPAR.LE.
     +    NPMAX .AND. NST.GT.0 .AND. NST.LE.NSTMAX) THEN
          READ (NIN,*) (PAR(I),I=1,NPAR), C
          READ (NIN,*) (ST(I),I=1,NST)
          READ (NIN,*) (ANV(I),I=1,NNV)
          IFAIL = 0
*
          CALL G13AGF(ST,NST,MR,PAR,NPAR,C,ANV,NNV,ANVR,WA,NWA,IFAIL)
*
          WRITE (NOUT,*)
          WRITE (NOUT,*)
     +      'The updated state set array now holds the values'
          WRITE (NOUT,99999) (ST(I),I=1,NST)
          WRITE (NOUT,*)
          WRITE (NOUT,99998) 'The residuals corresponding to the', NNV
          WRITE (NOUT,*) 'values used to update the system are'
          WRITE (NOUT,99999) (ANVR(I),I=1,NNV)
        END IF
        STOP
*
99999 FORMAT (1X,8F8.4)
99998 FORMAT (1X,A,I3,A)
        END
```

9.2   Program Data

```
G13AGF Example Program Data
  12
   0   1   1   0   1   1  12
  0.3270  0.6270  0.0000
  0.0118 -0.0669  0.1296 -0.0394  0.0422  0.1809  0.1211  0.0281
 -0.2231 -0.1181 -0.1468  0.0835  5.8201 -0.0157 -0.0361 -0.0266
 -0.0199  0.0298  0.0290  0.0147  0.0373 -0.0931  0.0223 -0.0172
 -0.0353 -0.0413
  5.8861  5.8348  6.0064  5.9814  6.0403  6.1570  6.3063  6.3261
  6.1377  6.0088  5.8916  6.0039
```

9.3   Program Results

```
G13AGF Example Program Results

The updated state set array now holds the values
  0.0660 -0.0513  0.1716 -0.0250  0.0589  0.1167  0.1493  0.0198
 -0.1884 -0.1289 -0.1172  0.1123  6.0039  0.0444 -0.0070  0.0253
  0.0019  0.0354 -0.0460  0.0374  0.0151 -0.0237  0.0032  0.0188
  0.0067  0.0126

The residuals corresponding to the 12
values used to update the system are
  0.0309  0.0031  0.0263  0.0105  0.0388 -0.0333  0.0265  0.0238
 -0.0159 -0.0020  0.0182  0.0126
```